

1. (Currently Amended) In a computer system providing a set of software system services, a method of providing replacement functions for the set of software system services, comprising:

 sending a request for a primitive function from one of the set of software system services to another one of the set of software system services, the primitive function replicating the another one of the set of software system services receiving the request for the primitive function in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error; and

 receiving an identifier associated with the requested primitive function at the one of the set of software system services sending the request for the primitive function from another one of the set of software system services, thereby enabling the one of the set of software system services to call the primitive function via the identifier associated with the requested primitive function instead of the another one of the set of software system services.

2. (Currently Amended) The method as recited in claim 1, wherein ~~sending a request for a primitive function and~~ receiving the identifier associated with the requested primitive function is performed only when the another one of the set of software system services receiving the request for the primitive function performs a debugging function.

3. (Currently Amended) The method as recited in claim 1, wherein ~~sending a request for a primitive function and~~ receiving the identifier associated with the requested primitive function is performed only when the another one of the set of software system services receiving the request for the primitive function performs at least one of an input and an output function.

4. (Currently Amended) A method of providing replacement functions for a stack of software system services, the stack of software system services including one or more layers, each layer representing one of the software system services, wherein lower layers provide services to upper layers in the stack, the method comprising:

 sending a primitive function request for a primitive function down from one of the layers of the stack of software system services to another one of the layers in the stack of

software system services, the primitive function replicating the system service associated with the another one of the layers in the stack receiving the primitive function request;

when the another one of the layers receiving the primitive function request is responsible for performing at least one of input and output, returning a primitive function identifier associated with the primitive function to the one of the layers of the stack of software system services sending the primitive function request.

5. (Currently Amended) The method as recited in claim 4, further comprising:

when the another one of the layers is responsible for performing at least one of input and output, sending another primitive function request from the another one of the layers in the stack of software system services that previously received the primitive function request from the one of layers of the stack of system services sending the primitive function request to a lower layer in the stack of software system services.

6. (Previously Amended) The method as recited in claim 4, further comprising:
propagating the primitive function request down the one or more layers of the stack of software system services.

7. (Currently Amended) In a computer system, a method of providing replacement functions for a stack of software system services, the stack of system services including one or more layers, each layer representing one of the software system services, wherein lower layers provide services to upper layers in the stack, the method comprising:

sending a primitive function request for a primitive function down from a first one of the layers in the stack of software system services to a second one of the layers in the stack of software system services, the primitive function replicating the system service associated with the second one of the layers in the stack of software system services in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error;

returning primitive function information associated with the primitive function to the first one of the layers sending the primitive function request; and

storing the primitive function information to enable the first one of the layers in the stack of system services sending the primitive function request to communicate with the primitive function associated with the second one of the layers in the stack of software

system services.

8. (Original) The method as recited in claim 7, wherein the primitive function information includes a pointer to the primitive function.

9. (Original) The method as recited in claim 7, wherein the primitive function information includes state information data to be provided to the primitive function when the primitive function is called.

10. (Previously Amended) The method as recited in claim 7, further comprising:
repeating the sending, returning, and storing steps over multiple layers of the stack such that a stack of primitive mechanisms parallel to the stack of software system services is assembled.

11. (Original) The method as recited in claim 7, wherein the returning and storing steps are performed when the second one of the layers in the stack contributes to at least one of input and output.

12. (Original) The method as recited in claim 7, wherein the returning and storing steps are performed when the second one of the layers in the stack contributes to debugging functions.

13. (Currently Amended) A system for providing replacement system functions in a computer system, comprising:

a set of software components providing a set of services;

a set of primitive software functions associated with the set of services, the set of primitive software functions replicating the set of services, wherein each of the set of primitive software functions eliminates or reduces reliance on one or more system functions that are capable of becoming non-functional in the event of a system error; and

a primitive function request mechanism adapted for being called by one of the set of software components providing the set of services and returning one or more identifiers associated with one or more of the set of primitive software functions to the one of the set of software components calling the primitive function request mechanism, thereby enabling the

one of the set of software components calling the primitive function request mechanism to call the one or more of the set of primitive software functions via the returned one or more identifiers.

14. (Previously Amended) The system as recited in claim 13, further comprising:
a primitive function calling mechanism adapted for calling one or more primitive software functions associated with the one or more identifiers returned by the primitive function request mechanism.

15. (Previously Amended) The system as recited in claim 14, wherein the primitive function calling mechanism is associated with one or more of the set of software components.

16. (Previously Amended) The system as recited in claim 13, wherein the one or more of the set of primitive software functions replace one or more of the set of services when the set of services are determined to be inoperative.

17. (Previously Amended) The system as recited in claim 13, wherein the one or more identifiers associated with one or more of the set of primitive software functions are returned in response to a primitive function request.

18. (Previously Amended) The system as recited in claim 13, further comprising:
state information associated with each of the set of software components, the state information including data that enables the corresponding service to communicate with another one of the set of services.

19. (Previously Amended) The system as recited in claim 13, further comprising:
state information associated with each of the set of components, the state information including data that enables the corresponding primitive software function to identify another one of the set of primitive software functions with which to communicate.

20. (Previously Amended) The system as recited in claim 13, wherein the set of services and the set of primitive software functions provide input and output functionality.

21. (Previously Amended) The system as recited in claim 13, wherein the set of services and the set of primitive software functions provide keyboard functionality.

22. (Currently Amended) A computer-readable medium for providing replacement functions for a set of software system services in a computer system, the computer-readable medium storing instructions thereon, comprising:

instructions for sending a request for a primitive function from one of the set of software system services to another one of the set of software system services, the primitive function replicating the another one of the set of software system services receiving the request for the primitive function in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error; and

instructions for receiving an identifier associated with the requested primitive function at the one of the set of software system services sending the request for a primitive function from another one of the set of software system services, thereby enabling the one of the set of software system services to call the primitive function via the identifier associated with the requested primitive function instead of the another one of the set of software system services.

23. (Currently Amended) An apparatus for providing replacement functions for a stack of software system services in a computer system, the stack of software system services including one or more layers, each layer representing one of the software system services, wherein lower layers provide services to upper layers in the stack, comprising:

means for sending a primitive function request for a primitive function down from a first one of the layers in the stack of software system services to a second one of the layers in the stack of software system services, the primitive function replicating the system service associated with the second one of the layers in the stack of software system services in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error;

means for returning primitive function information associated with the primitive function to the first one of the layers sending the primitive function request; and

means for storing the primitive function information to enable the first one of the layers in the stack of software system services sending the primitive function request to communicate with the primitive function associated with the second one of the layers in the stack of software system services.

24. (Previously Amended) The system as recited in claim 13, wherein the primitive function request mechanism is adapted for returning one or more identifiers associated with one or more of the set of primitive functions to one of the set of software system services in response to a primitive request sent by the one of the set of software system services.

25. (Previously Added) The method as recited in claim 1, wherein the set of software system services provide keyboard functionality.

26. (Previously Added) The method as recited in claim 1, wherein the set of software system services provide debugging functions.

27. (Previously Added) The method as recited in claim 1, wherein the set of software system services provides at least one of an input and an output function.

28. (Previously Added) The method as recited in claim 1, wherein the set of software system services provides input and output functionality.

29. (Previously Added) The method as recited in claim 1, wherein the primitive function uses polling.

30. (Previously Added) The method as recited in claim 1, wherein the primitive function does not include interrupts.

31. (Previously Added) The method as recited in claim 1, wherein the primitive function includes delay loops.

32. (Previously Added) The method as recited in claim 1, wherein the primitive

function does not include timers.

33. (Previously Added) The method as recited in claim 1, wherein sending a request for a primitive function and receiving the identifier associated with the requested primitive function is performed when the one of the set of software system services provides I/O functionality.

34. (Previously Added) The method as recited in claim 7, wherein the stack of software system services provide keyboard functionality.

35. (Previously Added) The method as recited in claim 7, wherein the stack of software system services provide debugging functions.

36. (Previously Added) The method as recited in claim 7, wherein the stack of software system services provides at least one of an input and an output function.

37. (Previously Added) The method as recited in claim 7, wherein the stack of software system services provides input and output functionality.

38. (Previously Added) The method as recited in claim 7, wherein the primitive function uses polling.

39. (Previously Added) The method as recited in claim 7, wherein the primitive function does not include interrupts.

40. (Previously Added) The method as recited in claim 7, wherein the primitive function includes delay loops.

41. (Previously Added) The method as recited in claim 7, wherein the primitive function does not include timers.

42. (Previously Added) The method as recited in claim 7, wherein sending a primitive

function request for a primitive function is performed when the first one of the layers in the stack of software system services provides I/O functionality.

43. (Previously Added) The method as recited in claim 7, wherein sending a primitive function request for a primitive function is performed only when the first one of the layers in the stack of software system services performs a debugging function.

44. (Previously Added) The method as recited in claim 7, wherein sending a primitive function request for a primitive function is performed only when the first one of the layers in the stack of software system services performs at least one of an input and an output function.

45. (Previously Added) The system as recited in claim 13, wherein the set of services provide keyboard functionality.

46. (Previously Added) The system as recited in claim 13, wherein the set of services provide debugging functions.

47. (Previously Added) The system as recited in claim 13, wherein the set of services provides at least one of an input and an output function.

48. (Previously Added) The system as recited in claim 13, wherein the set of services provides input and output functionality.

49. (Previously Added) The system as recited in claim 13, wherein the set of primitive functions use polling.

50. (Previously Added) The system as recited in claim 13, wherein the set of primitive functions do not include interrupts.

51. (Previously Added) The system as recited in claim 13, wherein the set of primitive functions include delay loops.

52. (Previously Added) The system as recited in claim 13, wherein the set of primitive

functions do not include timers.